AIAA-93-2720

A Strategy for the Optimal Design of Nozzle Contours

Stephen L. Keeling
Calspan Corporation/AEDC Operations
Arnold Air Force Base, Tennessee

AIAA 28th Thermophysics Conference
July 6-9, 1993 / Orlando, FL

A Strategy for the Optimal Design of Nozzle Contours*

Stephen L. Keeling**
Calspan Corporation/AEDC Operations
740 Fourth Street
Arnold Air Force Base, TN 37389-6001

Abstract

A strategy is proposed and analyzed for the aerodynamic design of optimally contoured, high-enthalpy, hypersonic nozzles. The approach involves expressing the desired contour as an optimal *convex* combination of trial configurations. The methods used are given a firm theoretical foundation. This includes mathematical uniqueness results that show what exit conditions guarantee uniform flow in a neighborhood of the nozzle exit. Also, convergence results are verified for the design scheme. Based upon this theoretical foundation, a modular, robust, axisymmetric nozzle design code is implemented. This design package is used successfully to design a nozzle that accelerates a turbulent, viscous perfect gas to a uniform Mach Number 4.0 flow in a neighborhood of the exit plane.

Nomenclature

A	Matrix defined for the quadratic programming problem in Eq. (6)
b	Vector defined for the quadratic programming problem in Eq. (6)
c_j, c_j^l	Coefficients used to define nozzle contours, f and f^l
C^k	Space of functions with k continuous derivatives
e	Total energy per unit mass
$\boldsymbol{e}$	Vector with all unit components
f	Nozzle contour function
f^l	Approximation to an optimal nozzle contour obtained after l iterations
f_j	Trial configuration or basis function used to define a nozzle contour
h, h_0	Total enthalpy per unit mass and its constant value at a nozzle exit
L	Lagrangian function defined in Eq. (8)
$\mathcal{L}$	Lagrangian function defined in Eq. (11)
M, M_0	Mach Number and its constant value at a nozzle exit
$\boldsymbol{M}$	Mach Number distribution for a given contour
M_i	Component of $\boldsymbol{M}$ for the ith grid cell
$\boldsymbol{M}'$	(Fréchet) derivative of the Mach Number distribution with respect to contour
M_i'	Component of $\boldsymbol{M}'$ for the ith grid cell
p, p_0, p_∞	Local static, stagnation, and free-stream static pressure
$\boldsymbol{Q}$	Vector of conservation variables
s, s_0	Entropy per unit mass and its constant value at a nozzle exit
T, T_0, T_∞, T_{wall}	Local static, stagnation, free-stream static, and wall temperature
u, v, w, u_∞	Cartesian velocity components and free-stream velocity
U, V, W	Matrices used for the singular value decomposition of Eq. (9)
x, y, z	Cartesian coordinates
$\boldsymbol{x}$	Vector defined for the quadratic programming problem in Eq. (6)

* The research reported herein was performed by the Arnold Engineering Development Center (AEDC), Air Force Materiel Command. Work and analysis for this research were done by personnel of Calspan Corporation/AEDC Operations, technical services contractor for the AEDC aerospace flight dynamics facilities. Further reproduction is authorized to satisfy needs of the U. S. Government.

** Senior Engineer, Computational Fluid Dynamics Section.

x_{exit}, x_{thrt}	x-coordinates of the exit and throat planes
y_0, z_0	(y,z)-coordinates of the centerline of an arbitrary cone in a 3D nozzle
y_{exit}	Half the width of the inviscid flow region in the exit plane of a planar nozzle
r	2D vector-valued function defining a grid
r_{exit}	Radius of the inviscid flow region in the exit plane of an axisymmetric nozzle
r_{thrt}	Radius in the throat plane of an axisymmetric nozzle
r_0	Distance from (y_0, z_0) to the inviscid flow region edge in a 3D nozzle exit plane
ζ, ζ^*	Computational coordinate in the radial direction and its maximum
λ_0	Lagrange multiplier defined in Eqs. (8) and (11)
λ	Vector of Lagrange multipliers defined in Eq. (11)
ξ, ξ^*	Computational coordinate in the streamwise direction and its maximum
ϱ, ϱ_∞	Local and free-stream density
$\phi_0, \phi_1, \psi_0, \psi_1$	Blending functions used to define r

1 Introduction

This paper is concerned with the aerodynamic design of optimally contoured hypersonic nozzles. The focus is primarily on axisymmetric nozzles, but the approach has been extended for other nozzle types. Typically, the challenge in the optimal design of nozzles consists in the following. Usually, the nozzle length and exit dimensions are specified. Then, the optimization problem is to determine the nozzle contour which minimizes the deviation of exit flow quantities from desired values. In particular, the present paper is focussed on the design of nozzles which produce a uniform exit flow at a specified Mach Number.

Sivells[1] has implemented a classical nozzle design strategy which can be described as follows. First, an inviscid flow contour is determined using the method of characteristics. Then this contour is corrected with a displacement thickness obtained from a boundary-layer calculation. Unfortunately, the flow quality in nozzles designed by Sivells' technique can be inadequate for high Mach Number, high-enthalpy flows. Also, such flows must be modeled more accurately, for example, by accounting for nonequilibrium effects. Therefore, more recently developed design approaches involve an iterative procedure in which adjustments are made to a prospective contour based on an accurate CFD (computational fluid dynamics) calculation of the nozzle flow field for that contour.

Korte, et al.[2] have successfully developed a CFD-based optimization code for the optimal design of nozzles supporting 2D hypersonic viscous flow. In Korte's approach the nozzle contour is determined by a smooth combination of cubic splines defined over a pre-established set of nodes. The nodal slopes of these piecewise polynomials are calculated iteratively by minimizing a certain objective function. Specifically, this function quantifies the deviation of the Mach Number from a specified value along part of the centerline and part of the exit plane. In addition, it is defined in terms of crossflow velocity profiles to minimize nonaxial velocity components near the exit. To compute these flow variables for the objective function, a CFD code is used which solves the parabolized Navier-Stokes equations. However, note that Korte's procedure requires a flow-field calculation for each entry in the Jacobian of the objective function with respect to design parameters. As explained in Section 4, such calculations are not required for the approach presented in this paper.

The present approach is related to that used by Barger and Moitra[3] to design aerodynamic bodies which closely match specified performance indices. In their work, the body shape is determined by a linear combination of trial configurations, each expected to be close to the optimal design. Then the optimal Fourier coefficients of this combination are calculated in a direct fashion. Specifically, this noniterative method involves a Gram-Schmidt orthogonalization procedure which minimizes the deviation of performance indices from specified values. It is intended that greater accuracy be achieved by increasing the number of trial configurations involved in the design. However, the stability of such a sequence of calculations cannot generally be guaranteed. As discussed in Section 6, such instabilities are guaranteed not to occur for the approach presented in this paper.

In the approach presented here, a hypersonic nozzle contour is designed as an optimal *convex* combination of prospective contours or basis functions. Specifically, a convex combination is defined as a linear combination in which the coefficients are non-negative and sum to one. As discussed in Sections 4 and 6, the consequences of constraining the combination to be convex are extremely significant.

The basis functions used to form possible contours are determined as follows. First, endpoint conditions and derivative bounds are specified for any possible contour. This defines a *solution set* which is convex; i.e., given any collection of curves in the set, every convex combination of the curves is also in the set. Also, there are *extreme* curves in the set which cannot be written as a convex combination of any other curves in the set. These are important since every possible solution curve can be approximated arbitrarily well with a convex combination of the extreme curves. Therefore, the basis functions are chosen from the collection of these extreme curves. In a similar fashion, basis functions can be chosen as perturbations of a baseline contour.

Once a set of basis functions is generated as described above, the associated nozzle flow field is computed for each. To be accepted, the set of contours must have the following properties. First, among the associated flow fields, there must be at least one Mach Number distribution that exceeds the specified value at every point in an inviscid flow neighborhood of the exit plane. Also, there must be at least one that is exceeded by the specified value. Finally, all flow fields must be smooth and, in particular, shock-free.

Once these preparations are completed, an iterative process begins with the selection of a random initial convex combination of the basis functions. After the flow field associated with this initial contour is computed, the next iterate must be determined. The intention is to choose the next iterate to minimize the deviation of the Mach Number from a specified value at certain points. These points can be chosen to drive the Mach Number to a constant along part of the centerline and part of the exit plane. Alternatively, the Mach Number can be driven to a constant, and its axial derivative to zero in part of the exit plane. As explained in Section 5, such conditions lead to uniform flow in a certain neighborhood of the nozzle exit.

The subsequent iterate is determined explicitly as follows. First, consider the nonlinear map which, for a given nozzle contour, gives the Mach Number at points situated near the exit plane as described above. This map is linearized with respect to the current iterate. As shown in Section 4, it is remarkable that the linearization can be approximated without referring explicitly to the (Frechet) derivative[4] of the map. This is a consequence of the fact that the Fourier coefficients of each iterate must sum to one. Finally, the coefficients of the next iterate are determined by minimizing, in a least-squares sense, the difference between the linearized Mach Number distribution and the desired Mach Number. Unfortunately, it has been found that if this least-squares problem is solved without constraining the new coefficients to be nonnegative, the sequence of iterates can diverge. On the other hand, if the convexity condition is enforced, the iterates converge rapidly. Thus, the coefficients of each new iterate are determined not as the solution to an unconstrained least-squares problem, but as the solution to a quadratic programming problem.

Finally, as discussed in Section 6, it is an important consequence of the convexity constraint that certain theoretical convergence results can be verified. Specifically, for a fixed number of basis functions, the iterative approximations are guaranteed to converge to a contour in the solution set. Also, any solution is guaranteed to be approximated arbitrarily well with a convex combination of a sufficiently large number of basis functions. Further, a sequence of optimal contours obtained with an ever-increasing number of basis functions is guaranteed to converge to a contour in the solution set.

The procedure outlined above is implemented in a modular fashion with three major modules at the highest level. First, for a given prospective contour, a nozzle grid is generated algebraically using Hermite transfinite interolation. This grid generator is discussed in Section 2. Next, the nozzle flow is calculated with a space-marching code due to Molvik and Merkle.[5] This is described in Section 3. After the Mach Number distribution is obtained from the flow solver, it is used in an optimization module to determine an improved contour. The quadratic programming techniques used in this module are explained in Section 4. Finally, computational results obtained using the complete nozzle design package are reported in Section 7.

2 Grid Generation

The requirements for the grid generation component of the nozzle design package are as follows. First, it is necessary to generate a grid for every prospective nozzle determined during a potentially lengthy iterative procedure. Therefore, these grids must be generated quickly. This need for speed suggests the use of an algebraic, as opposed to an elliptic approach. Also, the grid generator must operate robustly so that inputs need not be changed between iterations. Next, for the sake of modularity it is required that the grid generator interface smoothly with other codes, particularly flow solvers. Thus, a particular requirement is specified since many flow solvers use a conditionally correct method for imposing inviscid solid wall boundary conditions. Specifically, a method involving contravariant

velocity components has been found to give correct results only in case grid lines are orthogonal at the wall. (The details of this finding are omitted to limit the scope of this paper.) Thus, for inviscid flows, this orthogonality condition is a requirement for the grid generator. The orthogonality condition is also a requirement for viscous flows since certain turbulence models are formulated in terms of the distance from a solid wall. In addition, it is useful to have grid lines oriented normal to solid walls for convenient *a posteriori* boundary-layer analysis.

Well-established grid generation codes, such as the GRAPE code[6] and the EAGLE code[7] were options for the nozzle design package. However, to meet the requirements discussed above, it was necessary to implement an algebraic grid generation method based on Hermite transfinite interpolation.[8] Specifically, the grid is given by setting integer arguments in the vector-valued function:

$$r(\xi,\zeta) = \phi_0(\zeta)r(\xi,1) + \psi_0(\zeta)r_\zeta(\xi,1) + \phi_1(\zeta)r(\xi,\zeta^*) + \psi_1(\zeta)r_\zeta(\xi,\zeta^*)$$

$$1 \le \xi \le \xi^*, \qquad 1 \le \zeta \le \zeta^*$$

Here, ξ and ζ are computational coordinates increasing in the streamwise and radial directions, respectively. Also, the blending functions, ϕ_0, ϕ_1, ψ_0, ψ_1 are defined so that

$$\phi_0(1) = 1 \quad \phi_0(\zeta^*) = 0 \quad \phi_0'(1) = 0 \quad \phi_0'(\zeta^*) = 0$$

$$\phi_1(1) = 0 \quad \phi_1(\zeta^*) = 1 \quad \phi_1'(1) = 0 \quad \phi_1'(\zeta^*) = 0$$

$$\psi_0(1) = 0 \quad \psi_0(\zeta^*) = 0 \quad \psi_0'(1) = 1 \quad \psi_0'(\zeta^*) = 0$$

$$\psi_1(1) = 0 \quad \psi_1(\zeta^*) = 0 \quad \psi_1'(1) = 0 \quad \psi_1'(\zeta^*) = 1$$

Thus, $r(\xi, 1)$ is a boundary condition specifying the nozzle centerline, and $r(\xi, \zeta^*)$ specifies the nozzle contour. Also, $r_\zeta(\xi,1)$ and $r_\zeta(\xi, \zeta^*)$ are specified so that grid lines are orthogonal at the nozzle wall and centerline.

Finally, grid spacing is specified in the radial direction by defining the blending functions in terms of a certain distribution function, $s(t)$, $0 \le t \le 1$. Specifically, s is a smooth monotone function satisfying a power law at one extreme and increasing linearly at the other. Also, spacing is specified in the streamwise direction by setting $r(\xi,1)$ and $r(\xi,\zeta^*)$ in terms of a similar distribution function.

3 Flow Calculation

The requirements for the flow-solving component of the nozzle design package are as follows. First, it is necessary to compute the steady-state flow through every prospective nozzle determined during a potentially lengthy iterative procedure. Therefore, a time-iterative or pseudotime-iterative procedure is prohibitively time-consuming. Thus, it is necessary to space march through supersonic flow. Next, the flow solver must be able to compute complex flows through general 3D nozzles to solve design problems outside the scope of the present paper. In particular, the option must be available to compute the flow of a gas which is inviscid or viscous, laminar or turbulent. Also, it should be possible to model the fluid as a perfect gas, a gas in thermodynamic equilibrium, or a gas in chemical nonequilibrium. The flow solver must allow the specification of a range of boundary conditions. These include arbitrary supersonic inflow conditions, extrapolated outflow conditions, and tangent flow conditions at the centerline. At the nozzle wall, it must permit tangent flow or no-slip conditions and isothermal or adiabatic conditions. Furthermore, the code must operate robustly so that inputs need not be changed between iterations.

The requirements described above are met by the flow solver due to Molvik and Merkle.[5] Their code uses a conservative finite-volume space-marching technique to solve the chemically reacting parabolized Navier-Stokes equations. The difference scheme augments a first-order upwind Roe scheme[9] with higher-order terms which are inactivated in nonsmooth regions of flow. This numerical technique is due to Chakravarthy and Osher[10] and is designed to make the scheme total variation diminishing. Also, the code space marches the species continuity equations and the fluid dynamics equations, simultaneously, in strongly coupled form. The numerical method is fully implicit, so the nonlinear algebraic space-stepping equations are solved iteratively at each step. Specifically, the solution is obtained using a quasi-Newton iteration, due to Rai,[11] in which the Jacobian of the residual is simplified by approximate factorization. Continued quasi-Newton iterations eliminate linearization and factorization errors. Finally, stable marching through the subsonic boundary layer is accomplished by an approximation, due to Vigneron, et al.,[12] where the pressure in the streamwise flux is scaled so that streamwise growth modes admitted by the parabolic equations are eliminated.

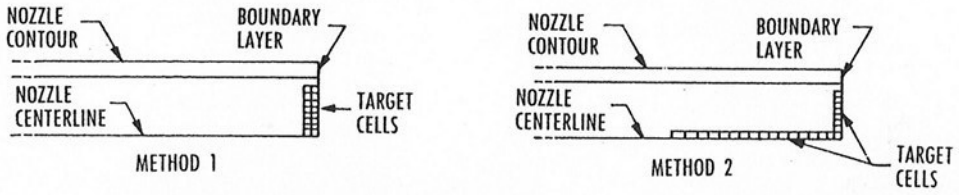

Fig. 1. Cells targeted for constant Mach Number to achieve uniform exit flow.

After the nozzle flow is computed, Mach Number values are extracted for use in the optimization scheme. As explained in Section 5, there are two natural approaches for achieving uniform exit flow at a specified Mach Number. Both methods are illustrated in Fig. 1. One method is to drive the Mach Number to a constant and its axial derivative to zero in an inviscid flow portion of the exit plane. To impose these conditions, Mach Number values are calculated for the two layers of cells nearest the nozzle exit, but within the inviscid flow region. Alternatively, the Mach Number can be driven to a constant in an inviscid flow portion of the exit plane and upstream from the exit along part of the centerline. Specifically, the ratio of the length of the centerline segment to the radius of the exit plane region is $\sqrt{M_0^2 - 1}$, where M_0 is the desired Mach Number. To impose these conditions, Mach Number values are calculated for appropriate cells along the exit plane and along the centerline.

4 Optimization

The purpose of the optimization procedure is to determine a nozzle contour for which the exit flow is uniform at a specified Mach Number, say M_0. For this, let $M(f)$ refer to the exit Mach Number distribution for a given nozzle contour, f. Specifically, let $M_i(f)$ denote the Mach Number in the ith grid cell taken from a collection of cells as discussed in the last section. Thus, the goal is to find a contour, f, from some set of functions, which minimizes the residual,

$$\sum_i |M_i(f) - M_0|^2. \qquad (1)$$

This nonlinear optimization problem is solved by an iterative procedure described below.

First, consider the set of admissible contours, or solution set. It is defined in terms of physically intuitive constraints. For example, a solution must be increasing and hence have a non-negative derivative. Also, a solution must have continuous curvature, and a certain number of its derivatives should be bounded. As explained in Section 1, such constraints lead to a convex solution set. Also, as discussed earlier, a curve in a convex solution set can be approximated with a convex combination of extreme curves, i.e., curves which cannot be written as a convex combination of any other curves in the set. Therefore, in the optimization scheme, a prospective contour, f, is a convex combination,

$$f(x) = \sum_j c_j f_j(x) \qquad \sum_j c_j = 1, \quad c_j \geq 0, \qquad (2)$$

where each basis function f_j is extreme with respect to the specified solution set. Note that if each basis function satisfies a certain equality or inequality constraint, the same holds for f. For example, if each f_j is increasing, so is f since

$$c_j \geq 0,\ f_j'(x) \geq 0 \text{ imply that } f'(x) = \sum_j c_j f_j'(x) \geq 0.$$

Now consider the explicit calculation of the basis functions. For example, suppose that the solution set is defined as follows. Let an exit radius and a range of throat radii be specified. Say that each permissible contour is required to have two continuous derivatives, and that the first derivative is non-negative. Assume that certain derivatives are specified at the exit and at the throat. Finally, suppose the third derivative (not necessarily continuous) is increasing between specified bounds. Then the extreme elements of the solution set happen to coincide with certain cubic spline functions described below. Therefore, these spline functions are used as basis functions. If a similar solution set is defined to require four continuous derivatives, the extreme elements are certain quintic spline functions.

There are many optional functional constraints to impose. The important point here is that these functional constraints translate to linear algebraic constraints on the coefficients of the piecewise polynomials defining a given

basis function. Now, suppose the number of spline nodes to be fixed and that the total number of independent polynomial coefficients is N. Each linear algebraic constraint determines a subset of the N-dimensional coefficient space, and only the points in the intersection of these subsets correspond to coefficients satisfying all constraints. Note that the intersection set is a convex hyperpolyhedron whose vertices cannot be expressed as a convex combination of any other points in the set. Also, every point in the set can be written as a convex combination of the vertices, i.e., the extreme points. The collection of spline functions corresponding to the points in this hyperpolyhedron is a discrete approximation to the complete solution set; however, the vertices correspond to functions which are actually extreme with respect to the solution set. Therefore, the basis functions are determined by choosing their polynomial coefficients to correspond to the vertices. These coefficients are calculated explicitly using a linear programming technique to find the active constraints on the boundary of the hyperpolyhedron. To limit the scope of this paper, the specifics of the linear programming technique are not given here, except for a reference to the basics appearing in Ref. 13.

It is remarkable that the representation shown in Eq. (2) allows the optimization to be formulated without explicit reference to the (Fréchet) derivative[4] of $M(f)$. Specifically, let the nozzle contour calculated at the lth iteration be denoted by

$$f^l(x) = \sum_j c_j^l f_j(x).$$

Then, f^{l+1} is computed from f^l by minimizing the linear approximation,

$$M_i(f^{l+1}) - M_0 \approx M_i(f^l) + M_i'(f^l)[f^{l+1} - f^l] - M_0 \tag{3}$$

in a least-squares sense. Here, M_i' is the (Fréchet) derivative of M_i. Its existence depends crucially upon the assumption that for every permissible contour, there is a well-defined, smooth flow field that depends smoothly upon the choice of contour.[14,15] Since $M_i'(f^l)$ is a linear operation on the bracketed expression,

$$\begin{aligned} M_i'(f^l)[f^{l+1} - f^l] &= \sum_j [c_j^{l+1} - c_j^l] M_i'(f^l) f_j \\ &= \sum_j [c_j^{l+1} - c_j^l] M_i'(f^l)[f_j - f^l]. \end{aligned} \tag{4}$$

The last equality follows since both $\{c_j^{l+1}\}$ and $\{c_j^l\}$ sum to one, and the added term, $M_i'(f^l)f^l$, does not depend upon j. Next, using the linear approximation,

$$M_i(f_j) \approx M_i(f^l) + M_i'(f^l)[f_j - f^l]$$

it follows that

$$\begin{aligned} &\sum_j [c_j^{l+1} - c_j^l] M_i'(f^l)[f_j - f^l] \\ &\quad \approx \sum_j [c_j^{l+1} - c_j^l][M_i(f_j) - M_i(f^l)] \\ &\quad = \sum_j [c_j^{l+1} - c_j^l] M_i(f_j). \end{aligned} \tag{5}$$

Again, the last equality follows since both sets of coefficients sum to one, and the extra term, $M_i(f^l)$, does not depend upon j. Now, combining Eqs. (3), (4) and (5) gives

$$M_i(f^{l+1}) - M_0 \approx M_i(f^l) - M_0 + \sum_j [c_j^{l+1} - c_j^l] M_i(f_j)$$

which does not depend explicitly upon the (Fréchet) derivative of $M(f)$. Thus, f^{l+1} is obtained by solving the quadratic programming problem

$$\begin{cases} \text{minimize } \|Ax - b\|^2 \\ \text{subject to } \sum_j x_j = 1, \quad x_j \geq 0 \end{cases} \tag{6}$$

where

$$A_{ij} = M_i(f_j), \qquad x_j = c_j^{l+1}$$

$$b_i = -M_i(f^l) + M_0 + \sum_j c_j^l M_i(f_j)$$

and setting

$$f^{l+1}(x) = \sum_j c_j^{l+1} f_j(x). \tag{7}$$

This quadratic programming problem is solved using a simplex-type method where heuristic rules are used to add or delete active constraints. Specifically, the minimization problem is first solved, unconstrained by the non-negativity

conditions. This is accomplished by finding a stationary point for the Lagrangian function,

$$L(x,\lambda_0) = \|Ax - b\|^2 - \lambda_0(e^t x - 1) \qquad (8)$$

where e has all unit components. Also, the superscript, t, denotes transposition. To account for the possibility that A is not well-conditioned, it is expressed in terms of its singular value decomposition[13]

$$A = UWV^t. \qquad (9)$$

Here, V is a square orthogonal matrix, i.e., $V^{-1} = V^t$. Its dimension is equal to the number of columns of A. Also, W is a diagonal matrix with the same dimension as V. Finally, U has the dimensions of A and has orthogonal columns. With this decomposition, the solution to the unconstrained minimization problem is given by

$$x = VW^{-1}U^t b + \frac{1 - e^t VW^{-1}U^t b}{e^t VW^{-2}V^t e} VW^{-2}V^t e. \qquad (10)$$

Unfortunately, this formula may yield some negative components for x. If so, the solution to the quadratic programming problem is situated on the boundary of the set $\{x : e^t x = 1, x_j \geq 0\}$, rather than in its interior. This point is found by locating a stationary point for the Lagrangian function,

$$\mathcal{L}(x,\lambda_0,\lambda) = \|Ax - b\|^2 - \lambda_0(e^t x - 1) - \lambda^t x \qquad (11)$$

with certain components of λ held fixed at zero. Specifically, it follows from the Kuhn-Tucker conditions for optimality,[16] that the solution is achieved at a point which is stationary with respect to the nonfixed arguments of $\mathcal{L}$ for which

$$x_j \geq 0, \qquad \lambda_j \geq 0, \qquad \lambda^t x = 0. \qquad (12)$$

This means that the solution has the property that the only components of λ which are nonzero are those for which the corresponding components of x are zero, and vice versa. Thus, the quadratic program solver proceeds as follows. Suppose that the kth component of x in Eq. (10) is negative and is actually the most negative among negative components. This suggests that the kth component of x be set to zero, so the kth component of λ becomes a free variable in Eq. (11). This can be implemented computationally by removing the kth column of A. Then, the solution is recalculated by Eq. (10) with the singular value decomposition in Eq. (9) corresponding to the new A. Continuing in this way, a non-negative x is eventually computed. Then, λ is obtained according to

$$\lambda = 2A^t(Ax - b) - 2\left[x^t A^t(Ax - b)\right]e \qquad (13)$$

where A has certain columns removed depending upon the path leading to the non-negative solution in Eq. (10). Suppose the kth component of λ in Eq. (13) is negative and is actually the most negative among negative components. This suggests that the kth component of λ be set to zero, so the kth component of x becomes a free variable in Eq. (11). This can be implemented computationally by restoring the kth column of A. Then, the solution is recalculated by Eq. (10) with the singular value decomposition in Eq. (9) corresponding to the new A. If this causes the new x to have negative components, they are eliminated according to the procedure outlined above. Again, λ is recomputed according to Eq. (13). Continuing in this way, the minimum point is eventually computed satisfying Eq. (12).

Since there are only finitely many inflection points to be tested, the method is guaranteed to converge provided it does not cycle. However, cycling is rare, and can be avoided easily with heuristic techniques.[16] In practice, this method for solving Eq. (6) converges rapidly without visiting a number of inflection points of combinatorial order. In fact, the quadratic program solver converges within a number of iterations on the order of the number of basis functions.

Once the quadratic programming problem of Eq. (6) is solved, f^{l+1} is determined by Eq. (7). Finally, this iteration on l continues until the residual in Eq. (1) or the difference between the coefficients, $\{c^{l+1}\}$ and $\{c^l\}$, meets a certain tolerance. See Section 6 for a discussion of the convergence.

5 Uniqueness Results

As stated in the last section, the purpose of the optimization procedure is not merely to determine a nozzle contour for which the exit flow reaches a specified Mach Number. In addition, the flow must be uniform in the exit region; i.e., all flow variables must be constant in a neighborhood of the exit plane. One way to achieve this is to explicitly minimize the deviation of all independent flow variables, at all points of a test region, from specified uniform conditions. However, these uniform conditions can be obtained by performing the minimization over a much smaller set. Indeed, the computational challenge is to determine the smallest subset of a test region on which constant flow properties will guarantee uniform flow throughout the test region. Specifically, given any region,

there exists a subset of its boundary on which the uniform conditions can be specified to determine a unique, uniform solution to the hyperbolic equations modeling inviscid, steady-state flow through the region. The purpose of this section is to state such mathematical uniqueness results. These provide the theoretical basis for minimizing the residual in Eq. (1) to determine the optimal nozzle contour. Uniqueness results will be stated for planar, axisymmetric, and general 3D problems, and careful proofs of these statements are omitted to limit the scope of this paper.

To state the uniqueness results, the geometry will be described in terms of Cartesian coordinates (x, y, z), where x increases in the downstream direction along the central axis of the nozzle. Also, for the following, let ϱ, u, v, w and e represent the density, the Cartesian components of velocity, and the total energy per unit mass, respectively. Then, set $Q = [\varrho, \varrho u, \varrho v, \varrho w, \varrho e]$. In each of the cases discussed below, Q will be assumed to be a solution to the Euler equations, linearized with respect to the desired constant flow state.

First, consider the planar case. Suppose Q is a smooth solution to the linearized Euler equations in the wedge-shaped region defined by

$$(y_{\text{exit}} - |y|)\sqrt{M_0^2 - 1} \geq (x_{\text{exit}} - x). \tag{14}$$

Here x_{exit} is the x-coordinate of the exit plane, and $2y_{\text{exit}}$ is the width of the inviscid flow region in the exit plane. Also, M_0 is a specified Mach Number. More specifically, suppose that Q has no variation with respect to z and is symmetric with respect to the xz plane. Next, assume that the following boundary conditions hold for constants h_0 and s_0:

$$\begin{cases} M = M_0 \\ M_x = 0 \\ h = h_0 \\ s = s_0 \end{cases} \text{for } x = x_{\text{exit}},\ 0 \leq |y| \leq y_{\text{exit}} \tag{15}$$

where M, h, and s represent the Mach Number, total enthalpy per unit mass, and entropy per unit mass, respectively. Then, Q is constant throughout the wedge-shaped region of Eq. (14). Alternatively, it can be shown that the same result holds if the boundary conditions of Eq. (15) are replaced with

$$\begin{cases} M = M_0 \\ h = h_0 \text{ for } x = x_{\text{exit}},\ 0 \leq |y| \leq y_{\text{exit}} \\ s = s_0 \\ M = M_0 \text{ for } x_{\text{exit}} - y_{\text{exit}}\sqrt{M_0^2 - 1} \leq x \leq x_{\text{exit}}, \\ \qquad\qquad y = 0 \end{cases} \tag{16}$$

Therefore, the conditions of either Eq. (15) or Eq. (16) can be used to achieve uniformity in the planar flow case.

Next, consider the axisymmetric case. Suppose Q is a smooth solution to the linearized Euler equations in the cone-shaped region defined by

$$\left(r_{\text{exit}} - \sqrt{y^2 + z^2}\right)\sqrt{M_0^2 - 1} \geq (x_{\text{exit}} - x). \tag{17}$$

Here, r_{exit} is the radius of the inviscid flow region in the exit plane. More specifically, suppose that Q has no variation with respect to $\theta = \tan^{-1}(z/y)$. Next, assume that the following boundary conditions hold for constants h_0 and s_0:

$$\begin{cases} M = M_0 \\ M_x = 0 \\ h = h_0 \\ s = s_0 \end{cases} \text{for } x = x_{\text{exit}},\ 0 \leq \sqrt{y^2 + z^2} \leq r_{\text{exit}} \tag{18}$$

Then, Q is constant throughout the cone-shaped region of Eq. (17). Alternatively, it can be shown that the same result holds if the boundary conditions of Eq. (18) are replaced with:

$$\begin{cases} M = M_0 \\ h = h_0 \text{ for } x = x_{\text{exit}},\ 0 \leq \sqrt{y^2 + z^2} \leq r_{\text{exit}} \\ s = s_0 \\ M = M_0 \text{ for } x_{\text{exit}} - r_{\text{exit}}\sqrt{M_0^2 - 1} \leq x \leq x_{\text{exit}}, \\ \qquad\qquad y = z = 0 \end{cases} \tag{19}$$

Hence, the conditions of either Eq. (18) or Eq. (19) can be used to achieve uniformity in the axisymmetric flow case.

Finally, consider the general 3D case. For example, the nozzle could have a circular cross section at the throat and

a square cross section at the exit. Suppose Q is a smooth solution to the linearized Euler equations in any cone-shaped region defined by

$$\left(r_0 - \sqrt{(y - y_0)^2 + (z - z_0)^2}\right)\sqrt{M_0^2 - 1} \geq (x_{\text{exit}} - x). \tag{20}$$

Here, r_0 is the distance from an arbitrary point (y_0, z_0) to the edge of the inviscid flow region, in the exit plane. Next, assume that the following boundary conditions hold for constants h_0 and s_0:

$$\begin{cases} M = M_0 \\ M_x = 0 \\ h = h_0 \\ s = s_0 \\ w_y = v_z \\ 0 = (y - y_0)v + (z - z_0)w \end{cases} \begin{array}{l} \\ \text{for } x = x_{\text{exit}}, \\ \\ 0 \leq \sqrt{(y - y_0)^2 + (z - z_0)^2} \leq r_0 \\ \\ \text{for } x = x_{\text{exit}},\ \sqrt{(y - y_0)^2 + (z - z_0)^2} = r_0 \end{array} \tag{21}$$

Then, Q is constant throughout the arbitrary cone-shaped region of Eq. (20). Unfortunately, for the general 3D case, Eq. (21) cannot be relaxed in a manner similar to that shown in Eqs. (16) and (19). Specifically, if the condition on the axial derivative, M_x, is replaced with a specification of the Mach Number along the centerline of the cone, where $y = y_0$ and $z = z_0$, then uniqueness of the constant solution cannot be established. In fact, it is straightforward to demonstrate multiple solutions explicitly.

The boundary conditions shown in Eqs. (15), (16), (18), (19) and (21) suggest that for nozzle design, the performance function to be minimized should involve all variables listed. In particular, this includes not only the Mach Number, but also the total enthalpy and the entropy. However, these thermodynamic properties do not appear in the residual function of Eq. (1). For flows considered in this study, total enthalpy and entropy are expected to be constant. On the other hand, additional terms should be included in the performance function for more general flows. Finally, note that while the uniqueness results of this section give a minimum set of theoretical conditions for uniform flow, it may be computationally convenient to overspecify with a larger set of conditions.

6 Convergence Results

This section is concerned with the convergence of three processes. The first is of the most immediate importance. It involves the behavior of iterates of the optimization scheme with continuing iterations for a fixed number of basis functions. Then, the second pertains to the approximation properties of the basis functions. Specifically, it is important for any solution to be approximated arbitrarily well with a convex combination of a sufficiently large number of basis functions. Finally, the third has to do with the behavior of a sequence of optimal contours obtained with an ever-increasing number of basis functions. Convergence results for these processes are only described below since careful proofs of these statements are beyond the scope of this paper.

First, consider the convergence of the iterative approximations for a fixed number of basis functions. It is a standard feature of inverse problems that the solution set should be carefully restricted.[17] Specifically, it should have at least the property that any sequence in it has a convergent subsequence; i.e., every sequence must accumulate around one or more elements in the set. This property is known as compactness.[4] For now it is assumed that the solution set is spanned by a fixed number of basis functions. Hence, it is finite-dimensional. In addition, the convexity constraint makes the solution set closed and bounded. All such finite-dimensional sets are compact. Thus, the iterates of the design scheme must be clustered around one or more accumulation contours. Also, an accumulation or limit contour must be a convex combination of the basis functions and therefore possess properties prescribed for the solution set.

Now, consider the approximation of any element in the solution set with a convex combination of basis functions. Here, the solution set is not assumed to be finite-dimensional. It is a convex set of functions limited only by constraints such as endpoint conditions and derivative bounds. Also, it is important to note for the following that if a bound is imposed on, say, the $(k + 1)$st derivative, the solution set is compact in C^k, the space of functions with continuous derivatives.[18] The concern is that there be no parts of the solution set which are inaccessible or which cannot be approximated arbitrarily well with a finite convex combination of a sufficiently large number of basis functions. However, (by the Krein-Milman Theorem)[19] every compact, convex set in C^k can be approximated arbitrarily well (in a C^k sense) with convex combinations of its extreme elements. In this work, the basis functions are constucted to approximate the complete set of extreme elements of the solution set with arbitrary accuracy as the number of basis functions increases. Hence, convex combinations of them can be used to approximate any element of the solution set with arbitrary accuracy.

Finally, consider the behavior of a sequence of contours, each of which is optimal in the convex hull of a fixed number of basis functions. It is desirable for this sequence to converge with respect to an ever-increasing number of basis functions. Based upon discussions above, note that each optimal contour lies in a finite-dimensional approximation to the full infinite-dimensional solution set. Since the infinite-dimensional set contains all of its finite-dimensional approximations, the sequence of optimal contours in question must lie in the full infinite-dimensional solution set. Recall that the full set is compact in C^k if a bound is imposed on the $(k + 1)$st derivative. Thus, the sequence of optimal contours must cluster (in a C^k sense) around one or more accumulation contours. Also, such an accumulation contour must lie in the full solution set and therefore possess the properties prescribed for any admissible solution.

7 Computational Results

In this section, a test nozzle design calculation is presented. This example involves the optimal design of an axisymmetric nozzle through which a turbulent, viscous perfect gas (air) is accelerated to a uniform Mach Number 4.0 flow in a neighborhood of the exit plane.

The geometric constraints were:

$$x_{\text{exit}} - x_{\text{thrt}} = 25.0 \text{ in.}, \quad f(x_{\text{exit}}) = 2.22 \text{ in.},$$
$$0.5 \text{ in.} \le f(x_{\text{thrt}}) \le 0.7 \text{ in.},$$
$$f'(x_{\text{thrt}}) = 0, \qquad f'(x_{\text{exit}}) = 0.01, \qquad f''(x_{\text{exit}}) = 0$$
$$-0.08 \le f'''(x) \le 0.01, \qquad f''' \text{ increasing.}$$

With 14 equally spaced nodes, these conditions generated 20 cubic spline basis functions according to the procedure outlined in Section 4. They are shown in Fig. 2. Other contraints have been shown to work well except when $f'(x_{\text{exit}})$ is not positive. The value used here was suggested by an optimization where $f'(x_{\text{exit}})$ was left unspecified. As expected, the accuracy obtained was shown to increase as the number of nodes was increased from 7 to 14.

The grids generated for these basis functions and for optimization iterates contained 301 points in the streamwise direction, by 71 points in the radial direction. The viscous spacing near the wall satisfied a power law with an exponent of 9. Also, spacing was tighter near the throat, satisfying a power law with an exponent of 7/3. For space marching, the flow solver interpolated from such grids for a total of 559 space steps. The initial space step was 0.001

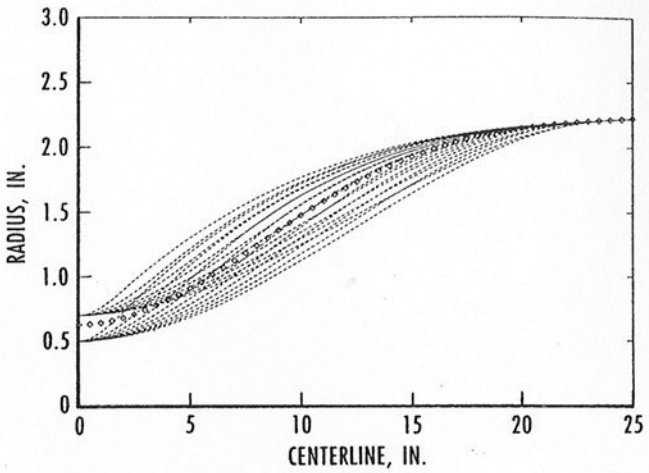

Fig. 2. Optimal contour (symbols) plotted among basis functions (dashed curves).

in., which was increased by a factor of 1.02 at each step to a maximum step of 0.2 in. In each case, the space marching was initialized with a calculation of flow through a cylinder extending smoothly upstream from the throat for a length of 0.01 in. Note that the throat location is understood as the point of initial expansion.

The stagnation conditions were

$$p_0 = 60 \text{ lb/in.}^2 \qquad T_0 = 630°\text{R}$$

The Reynolds Number was 1.26×10^6 for a 1-in. reference length. The flow variables were specified at the inflow boundary to give a constant inviscid core flow with a certain boundary-layer profile. Specifically, free-stream conditions were set in the inviscid core so that nonaxial velocity components were zero and all other variables were determined from a Mach Number of 1.01. For example,

$$u_\infty = 1.13 \times 10^3 \text{ ft/sec} \qquad T_\infty = 5.23 \times 10^2 \text{ °R}$$
$$p_\infty = 4.51 \times 10^3 \text{ lb/ft}^2 \qquad \varrho_\infty = 5.02 \times 10^{-3} \text{ slug/ft}^3$$

Then the boundary-layer profile was generated as follows:

$$\sqrt{\frac{T - T_{\text{wall}}}{T_\infty - T_{\text{wall}}}} = \frac{u}{u_\infty} =$$

$$\begin{cases} 1 & 0 \le r \le r_{\text{thrt}}(1 - \Delta) \\ \left(\dfrac{r_{\text{thrt}} - r}{r_{\text{thrt}}\Delta}\right)^{\frac{1}{q}} \left[q + 1 - \left(\dfrac{r_{\text{thrt}} - r}{r_{\text{thrt}}\Delta}\right)\right]\dfrac{1}{q} & r_{\text{thrt}}(1 - \Delta) \le r \le r_{\text{thrt}} \end{cases}$$

Here, $\Delta = 0.017$ and $q = 2.7$. Also, the wall temperature, T_{wall}, was set to 90 percent of the free-stream temperature, T_∞. The bracketed expression was used to make the profile smooth.

Besides this inflow profile, the following additional boundary conditions were imposed. At the centerline, a tangent flow condition was specified. At the nozzle wall, a viscous no-slip condition was imposed. Also, the wall temperature was specified to be constant and the same value as indicated for the inflow boundary. The flow solver was run using a Baldwin-Lomax algebraic turbulence model. The number of quasi-Newton iterations performed by the flow solver was set to two. For the optimization scheme, the cells used for the residual of Eq. (1) were taken from the exit plane and the centerline. The layer of cells along the exit plane ranged from the center to 70 percent of the exit radius. The layer of cells along the centerline was longer by a factor of $\sqrt{M_0^2 - 1} = \sqrt{15}$.

With these conditions in place, the optimization package required two iterations for the residual in Eq. (1) to drop below 10^{-3}. This calculation was completed on a Silicon Graphics IRIS Indigo® workstation, with each flow-field calculation requiring less than 200 sec. The final contour is shown among basis functions in Fig. 2. Also, a contour plot of the Mach Number distribution is shown in Fig. 3. Note the constant Mach Number region near the exit. In this region, the Mach Number deviates from 4.0 by less than 0.1 percent.

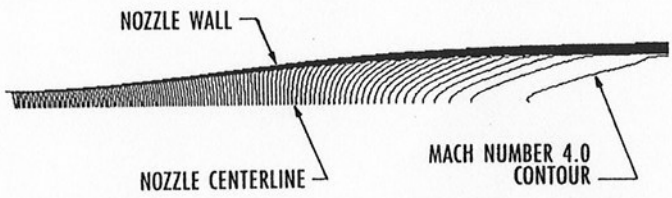

Fig. 3. Mach Number contour plot for a Mach Number 4.0 nozzle.

8 Concluding Remarks

A strategy has been proposed and analyzed for the aerodynamic design of optimally contoured, high-enthalpy, hypersonic nozzles. The approach involves expressing the desired contour as an optimal convex combination of trial configurations. The methods used were given a firm theoretical foundation. This includes mathematical uniqueness results that show what exit conditions guarantee uniform flow in a neighborhood of the nozzle exit. Such results have been proved for planar, axisymmetric, and generally 3D nozzles. Also, convergence results were verified for the design scheme. Specifically, for a fixed number of basis functions, the iterative approximations are guaranteed to converge to a contour in the specified solution set. Also, any solution can be approximated arbitrarily well with a convex combination of a sufficiently large number of basis functions. Finally, a sequence of optimal contours obtained with an ever-increasing number of basis functions is guaranteed to converge to a contour in the solution set. After the theoretical foundation was secured, a modular, robust, axisymmetric nozzle design package was implemented. This package includes an algebraic grid generator module, a space-marching flow solver module, and a quadratic program solver module. The design package was used to design a nozzle that accelerates a turbulent, viscous perfect gas to a uniform exit flow that deviates from Mach Number 4.0 by less than 0.1 percent in a neighborhood of the exit plane. The quality of the computational results suggests that the design scheme will be successful in the design of more complex nozzles, supporting more complex flows. In fact, the optimizer strategy can be applied to other aerodynamic design problems.

References

1. Sivells, J. C., "A Computer Program for the Aerodynamic Design of Axisymmetric and Planar Nozzles for Supersonic and Hypersonic Wind Tunnels," AEDC-TR-78-63 (AD-A062944), December 1978.

2. Korte, J. J., Kumar, A., Singh, D. J. and White, J. A., "CAN-DO, CFD-Based Aerodynamic Nozzle Design and Optimization Program for Supersonic/Hypersonic Wind Tunnels," AIAA-92-4009, July 1992.

3. Barger, R. L. and Moitra, A., "A Performance Index Approach to Aerodynamic Design with the Use of Analysis Codes Only," NASA-TP-2905, 1988.

4. Hudson, V. and Pym, J. S., *Applications of Functional Analysis and Operator Theory*, Academic Press, New York, 1980, pp. 123, 139-140.

5. Molvik, G. A. and Merkle, C. L., "A Set of Strongly Coupled, Upwind Algorithms for Computing Flows in Chemical Nonequilibrium," AIAA-89-0199, January 1989.

6. Sorenson, R. L., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation," NASA-TM-81198, May 1980.

7. Thompson, J. F., Dietz, W. E., Thomson, W. G., Clippard, R. L. and Rist, M. J., "Interactive EAGLE: An Interactive Surface Mesh and Three-Dimensional Grid Generation System," AIAA-90-1569, June 1990.

8. Thompson, J. F., Warsi, Z. U. A. and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, North Holland, New York, 1985, p. 315.

9. Roe, P. L., Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, *Journal of Comparative Physics*, Vol. 43, 1981, pp. 357-372.

10. Chakravarthy, S. R., and Osher, S., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA-85-0363, 1985.

11. Rai, M. M., "An Implicit Form for the Osher Upwind Scheme," AIAA-84-0088, January 1984.

12. Vigneron, Y. C., Rakich, J. C. and Tannehill, J. C., "Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Subsonic Leading Edges," AIAA-78-1137, July 1978.

13. Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T., *Numerical Recipes*, The Art of Scientific Computing, Cambridge University Press, Cambridge, 1989, pp. 52-64, 312-326.

14. Solonnikov, V. A. and Kazhikhov, A. V. , Existence Theorems for the Equations of Motion of a Compressible Viscous Fluid , *Ann. Rev. Fluid Mech.*, Vol. 13, 1981, pp. 79-95.

15. Temam, R., *Navier-Stokes Equations, Theory and Numerical Analysis*, North-Holland, Amsterdam, 1984, pp. 20, 157, 278, 500.

16. Gill, P. E., Murray, W. and Wright, M. H., *Practical Optimization*, Academic Press, New York, 1981, pp. 67-74, 328.

17. Tikhonov, A. N. and Arsenin, V. Y., *Solutions of Ill-Posed Problems*, V. H. Winston and Sons, Washington D. C., 1977, pp. 9, 28-29.

18. Adams, R. A., *Sobolev Spaces*, Academic Press, New York, 1975, pp. 11, 97-98, 218-219.

19. Rudin, W., *Functional Analysis*, McGraw-Hill, New York, 1973, pp. 36, 70.